

基于用户评分和共同评分项的协同过滤算法研究 *

张 宏[†], 王 慧

(浙江理工大学 经济管理学院, 杭州 310018)

摘 要: 针对协同过滤算法存在的问题进行改进, 以提高评分预测和推荐结果的准确性。传统的相似度度量方法只考虑用户评分, 过于简单, 在皮尔逊相似度的基础上引入用户评分时间和商品流行度对用户评分进行加权处理, 并基于共同评分项规模的相似度计算进行加权组合, 使得计算结果更加准确, 也更符合现实意义。实验结果表明, 新算法评分预测的平均绝对误差明显低于皮尔逊相似度, 将 MAE 降低了 10% 以上, 并提高了推荐的召回率和覆盖率。只在电影评分数据集上进行实验验证有一定的局限, 该算法能够提高协同过滤算法的准确性, 具有一定的现实意义。

关键词: 协同过滤; 皮尔逊相似度; 评分时间; 共同评分项; 商品流行度

中图分类号: TP301.6 **doi:** 10.3969/j.issn.1001-3695.2017.07.0696

Research on collaborative filtering algorithm based on user score and common score

Zhang Hong[†], Wang Hui

(School of Economics & Management Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: In order to improve the accuracy of the scoring prediction and the proposed results, the algorithm is improved to solve the problems existing in the collaborative filtering algorithm. The traditional similarity measure method only considers the user score, which is too simple. Based on the Pearson correlation coefficient, this paper introduces the user's score time and commodity prevalence to weight the user's score, and then combine with the similarity calculation based on the scale of the common score. The new algorithm can make the calculation results more accurate, but also more realistic. The experimental results show that the average absolute error of the new algorithm score is obviously lower than that of Pearson correlation coefficient, and the MAE is reduced by 10%, and the recommended recall rate and coverage rate are also improved. [Limit] But the experiment is only carried out on movie rating data set, with some limitations. This algorithm can improve the accuracy of cooperative filtering algorithm and has certain practical significance.

Key Words: collaborative filtering; Pearson correlation coefficient; score time; co-rating; commodity heat

0 引言

信息技术与电子商务的发展极大地方便了人们的生活, 但是随之骤增的信息量, 也带来了信息过载问题, 致使消费者很难在大量的信息中找到对自己有用的信息。搜索引擎能够进行信息过滤, 但是不能提供“个性化”的服务, 个性化推荐技术的出现有效地缓解了信息过载给消费者带来的困扰, 它的核心是根据消费者的个人喜好向其推荐相关的产品。

个性化推荐技术有很多相关算法, 协同过滤是目前应用最为广泛的算法之一, 它于 1992 年由 Goldberg 等人^[1]提出。协同过滤分为两种类型: 基于模型的协同过滤和基于记忆的协同过滤^[2]。基于模型的协同过滤是从历史评分数据中学习得到相关模型, 然后利用模型进行评分预测。基于记忆的协同过滤分为 user-based CF 和 item-based CF^[2], 它们的主要思想是, 根据

用户(商品)相似度确定邻居用户, 然后利用邻居用户对商品的评分来预测用户对该商品的评分, 从而产生推荐列表。由此可见, 用户相似度度量的准确性对推荐效果有着显著的影响, 但是传统的相似度度量方法, 如余弦相似度、改进的余弦相似度、皮尔逊相似度等^[3-4], 仅考虑用户之间的共同评分的项目, 在用户评分矩阵稀疏的情况下, 其计算出的相似度准确性并不好。

为了进一步优化协同过滤算法, 许多相关领域的专家学者对提高相似度计算进行了研究, 主要分以下三个方向:

a)改进传统的相似度度量方法。Hao 等人^[5]利用平衡因子计算用户间的项目评分差异, 并将它加入到传统的余弦相似度算法中, 其中最优的平衡因子通过实验获得; 而 Liuet^[6]在计算用户相似度时不仅考虑用户的评分信息, 将用户行为信息也考虑入内; Ester 等人^[7]则基于 sigmoid 函数提出了一种新的相似

基金项目: 国家社会科学基金资助项目(15BSH107)

作者简介: 张宏(1978-), 女(通信作者), 教授, 博士, 主要研究方向为电子商务、个性化营销、战略管理(zhh96941@zju.edu.cn); 王慧(1993-), 女, 硕士研究生, 主要研究方向为电子商务、数据挖掘、个性化营销。

度量方法, 这种方法可以削弱用户间共同评分较低的商品对相似度计算的影响。

b) 对评分矩阵进行降维处理。Gong [8] 结合了奇异值分解 (SVD) [9] 和 user-based CF, 他利用奇异值分解的结果来填补缺失值, 然后用 user-based CF 产生推荐, 这种组合方法可以提高系统的精度。Pirasteh [10] 通过降低矩阵维度来产生推荐, 并且利用电影的类型和导演信息等额外的信息进行相似度计算。除此之外, 主成分分析 (PCA) 也是降维处理中常用的方法。

c) 将其他技术和协同过滤算法相结合。Tsai [11] 提出分阶段分数预测模型, 根据最初的评分矩阵, 使用聚类方法首先将用户和商品进行聚类, 对早期商品的聚类结果使用加权非负矩阵分解方法, 然后进行预测和推荐。Wang 等人 [12] 改善了数据分组的算法, 通过使用 Hamming 距离, 使得聚类的准确性更高, 然后使用 Slope 方法进行推荐。除此之外, 社会网络结构 [13]、数据挖掘 [14] 和机器学习 [15] 等方法也被应用到相似度量度中。

以上方法均在不同程度上对相似度计算进行了改进, 并且得到了一定的效果。但是以上几种方法和传统的相似度计算方法一样, 忽略了用户评分时间、共同评分项的规模和商品流行度的影响。本文针对皮尔逊相似度存在的问题, 引入用户评分时间和商品流行度两个因素对用户评分进行加权处理, 并与基于用户共同评分项规模的相似度算法进行结合, 得到新的相似度量度方法 RJpearson, 并在 Movielens 数据集进行实验验证。实验结果表明, 本文提出的方法能够有效降低评分预测的误差, 提高相似度量度的准确性。

1 相关工作

1.1 传统的协同过滤算法

协同过滤算法分为基于用户的协同过滤和基于项目的协同过滤。两者的主要步骤相似: 根据历史评分进行相似度量度、产生邻居列表、进行评分预测, 然后根据预测的评分产生列表。本文针对基于用户的协同过滤进行研究, 具体的算法流程如图 1 所示。

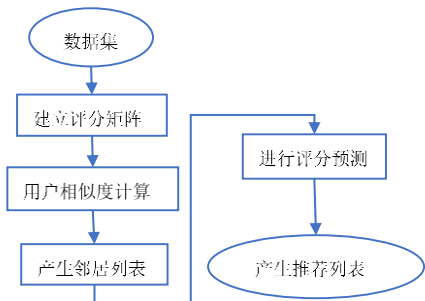


图 1 User-based CF 流程图

相似度量度是协同过滤算法的核心, 常用的相似度量度方法有余弦相似度、修正的余弦相似度和皮尔逊相似度等。其中皮尔逊相似度 (Pearson correlation coefficient) 计算两个变量间的线性相关关系, 公式如下:

$$Pearson(i, j) = \frac{\sum_{k \in I_{ij}} (r_{ik} - \bar{r}_i) \times (r_{jk} - \bar{r}_j)}{\sqrt{\sum_{k \in I_{ij}} (r_{ik} - \bar{r}_i)^2} \times \sqrt{\sum_{k \in I_{ij}} (r_{jk} - \bar{r}_j)^2}} \quad (1)$$

其中: $Pearson(i, j)$ 表示用户 i 和 j 的相似度, I_{ij} 表示用户 i 和 j 共同评分的商品集合, r_{ik} 和 r_{jk} 分别表示用户 i 和用户 j 对商品 k 的评分, \bar{r}_i 和 \bar{r}_j 分别表示用户 i 和用户 j 对商品的平均评分。

评分预测的公式定义如下:

$$P_{ik} = \bar{r}_i + \frac{\sum_{j \in I_i} Sim(i, j) \times (r_{jk} - \bar{r}_j)}{\sum_{j \in I_i} Sim(i, j)} \quad (2)$$

其中: \bar{r}_i 表示目标用户 i 对所有评分商品的评分均值, I_i 表示目标用户的邻居集合, r_{jk} 表示邻居用户 j 对商品 k 的评分, \bar{r}_j 表示邻居用户 j 对所有评分商品的评分均值, $Sim(i, j)$ 表示目标用户 i 与邻居用户 j 的相似度。协同过滤因其简单有效而得到广泛使用, 但同时也存在很多不足。

1.2 相似度量度存在的问题

1.2.1 忽略共同评分项的规模

电子商务平台中存在着大量的用户和商品, 但是每个用户所购买或评价的商品数量有限, 导致用户评分矩阵十分稀疏 [16], 即用户之间有可能存在很少的共同评分项。共同评分项的规模会对用户相似度的计算造成影响, 比如用户 user1 和 user2 对所购商品的评分如表 1 所示。

表 1 用户评分						
	I1	I2	I3	I4	I5	I6
U1	3	2	0	5	4	0
U2	0	2	3	0	0	4

使用 Pearson 相似度量度方法计算, U1 和 U2 的相似度为 1, 但实际上, 有评分矩阵明显看出, U1 偏好 I1、I4、I5, 而 U2 偏好 I3、I6, 两者的偏好行为并不相似。传统的 Pearson 相似度量度方法忽略共同评分项的规模, 导致相似度计算的准确性降低。

1.2.2 忽略用户评分时间的影响

时间标签是推荐系统的一个重要因素, 它能够体现用户的兴趣迁移。用户在不同的时间内, 兴趣爱好会发现很大的变化。例如, 夏季用户会对降温产品的关注度较高, 而冬季, 用户则会多保暖物品更有兴趣。但是传统的相似度量度方法却忽略了这一重要的因素, 而不能将用户兴趣爱好的变化反映在相似度量度中, 导致推荐的结果不一定符合用户当下的偏好。

1.2.3 忽略商品流行度的影响

用户在选购商品时往往存在从众心理, 所谓“从众”是指在群体的影响下放弃自己的主张而与大家保持一致的社会心理行为。如网购的时候, 用户会习惯性地看商品的销量, 销量很高时, 用户购买以及给高分的倾向就越大。因此商品的流行度

会对消费者的购买行为产生影响。Pearson 相似度计算用户相似度时忽略了商品流行度对相似度计算的影响, 一方面使得相似度度量准确性不够, 另一方面还会导致推荐结果大多是流行度较高的商品, 而忽略了对其他商品的推荐。真正的“个性化”推荐, 不仅要能做到向用户推荐热门的商品, 还要为用户推荐体现其偏好行为的商品。

针对以上问题, 为了提高用户相似度度量的准确性和推荐效果, 本文在 Pearson 相似度的基础上, 提出一种新的用户相似度度量方法, 在下一部分进行详细介绍。

2 基于用户评分和共同评分项的算法改进

2.1 基于用户评分的相似度计算

为了体现用户的当前喜好以及排除商品流行度对用户评分的影响, 在进行用户间 Pearson 相似度计算时, 首先加入惩罚因子和 logistics 时间权重函数对用户评分进行处理, 其次再利用式 (1) 进行相似度计算。

2.1.1 惩罚因子

为了解决 Pearson 相似度计算时忽略商品流行度的问题, 引入惩罚因子对用户评分进行加权处理。商品流行度用 N_k 衡量, N_k 表示对商品 k 进行评分的用户数, N_k 越大说明该商品越热门。为了减小商品流行度对用户相似度的影响, 将商品 K 惩罚因子的公式定义如下:

$$penalty(k) = \frac{1}{\log(1 + N_k)} \quad (3)$$

商品 k 得到的评价越多 (即商品越热门), 则对商品 k 产生的惩罚性越大, 在计算用户间相似度时, 该用户对商品 k 的评分产生的影响就越小。因此, 惩罚因子的引入可以削弱商品流行度对用户相似度计算造成的误差。

2.1.2 logistic 时间权重函数

为了解决传统算法不能反映出用户兴趣爱好随时间变化的问题, 本文在计算用户相似度时加入 logistic 时间权重函数^[17-18], 运用 logistic 函数对用户评分进行时间加权计算, 以达到增加近期评分的权重的目的, logistic 函数为:

$$f(t_{ik}) = \frac{1}{1 + e^{-t_{ik}}} \quad (4)$$

其中: t_{ik} 表示用户 i 对商品 k 的评分时间与该用户最早产生评分的时间之差。函数 $f(t_{ik})$ 的值随着 t_{ik} 的增加而呈非线性曲线增长, 即评分时间越新, 用户评分在相似度计算时产生的作用就越大。该函数的取值范围为 (0,1), 对用户评分进行时间加权处理改善了传统算法仅利用用户评分进行推荐的不足, 更好地为目标用户选择相似性更高的邻居用户。

2.1.3 基于用户评分的相似度计算

根据商品流行度和 logistic 时间权重函数对用户评分进行评分加权处理:

$$R_{ik} = r_{ik} * penalty_k * f(t_{ik}) \quad (5)$$

其中: r_{ik} 表示用户 i 对商品 k 的评分, $penalty_k$ 表示商品 k 的惩

罚因子, $f(t_{ik})$ 表示用 i 对商品 k 的评分时间的 logistic 时间权值。经过加权处理得到新的评分矩阵, 针对新的评分矩阵进行基于用户评分的相似度计算, 计算公式如下:

$$R_{pearson}(i, j) = \frac{\sum_{k \in I_{ij}} (R_{ik} - \bar{R}_i) * (R_{jk} - \bar{R}_j)}{\sqrt{\sum_{k \in I_{ij}} (R_{ik} - \bar{R}_i)^2} * \sqrt{\sum_{k \in I_{ij}} (R_{jk} - \bar{R}_j)^2}} \quad (6)$$

其中: $R_{Pearson}(i, j)$ 表示用户 i 和用户 j 的相似度, R_{ik} 和 R_{jk} 分别表示加权处理后用户 i 和用户 j 对商品 k 的评分, \bar{R}_i 和 \bar{R}_j 分别表示用户 i 和用户 j 的平均评分。

2.2 基于共同评分项的相似度计算

衡量共同评分项的规模一般采用 Jaccard 相似度度量方法, 计算公式定义如下:

$$Jaccard(i, j) = \frac{|I_i \cap I_j|}{|I_i \cup I_j|} \quad (7)$$

I_i 和 I_j 分别是用户 i 和用户 j 的评分集合, $Jaccard(i, j)$ 相似度利用两个用户评分集合的交集 (即共同评分项数目) 与两用户评分集合的并集的比值来衡量两个用户间的相似度。在两个用户的评分集中, 若两个用户的共同评分项越多, 则两用户的相似度越高, 即两个用户的兴趣爱好越接近; 反之, 亦然。Jaccard 相似度可以很好地反映用户之间基于共同评分项的相似度。

2.3 基于用户评分和共同评分项的算法改进

将基于用户评分的相似度 $R_{pearson}$ 和基于共同评分项的相似度 $Jaccard$ 进行融合, 得到 $RJ_{Pearson}$ 相似度。公式如下:

$$RJ_{Pearson}(i, j) = \lambda R_{pearson}(i, j) + (1 - \lambda) Jaccard(i, j) \quad (8)$$

其中: λ 为平衡参数, 取值范围为 (0,1), 根据不同的 λ 取值得到不同的 $RJ_{Pearson}$ 相似度, 改进后的算法流程如图 2 所示。

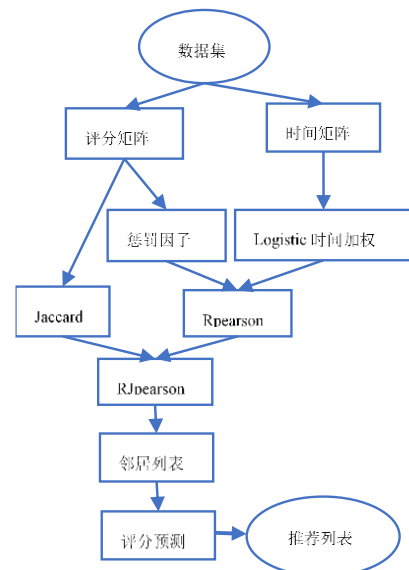


图2 RJpearson 算法流程图

3 实验

3.1 实验数据集

本实验采用的数据集是来自于 Grouplens 提供的 MoviesLens-100k 电影评分数据集。该数据集中评分制是 5 分制,它包含了 943 个用户和 1682 部电影,其中每个用户至少对 20 部以上的电影进行了评分。

从数据集中随机抽取 80%的样本数据作为训练集,其余的 20%作为测试集。在训练集上进行相似度的计算,并对测试集中的电影进行评分预测,从而与 test 数据集的实际评分进行对比,得出相关实验结果。

本实验的对比算法:传统的 use-based CF 算法和基于 Jaccard 相似度的推荐算法。

3.2 评价指标

在衡量推荐系统的质量时,评价标准有很多种,本文选取较常用的平均绝对误差法(MAE)和准确率召回率方法。

在比较实际值和预测值之间的误差时,MAE 方法是使用频率最高的评价标准,其计算公式如下:

MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N} (9)

其中, p_i 和 q_i 分别代表对用户的预测评分和用户的实际评分。从公式可知,预测评分跟实际评分越接近,两者之间误差的绝对值就越小,MAE 的值也就越小,说明预测效果就越好。

准确率召回率一般是用来衡量推荐算法返回的推荐列表的效果。准确率(precision)和召回率(recall)计算公式如下:

Precision = \frac{n}{R_n} (10)

Recall = \frac{n}{N} (11)

其中: n 表示正确推荐的商品数,即系统推荐的商品在用户实际购买的商品中出现的次数; N 表示用户真正需要推荐的商品数,即用户实际购买的商品数量; R_n 表示系统推荐的商品数。准确率和召回率往往是矛盾的,即当准确率提高时,召回率会下降,反之亦然。所以,为了综合考虑召回率和准确率,本文采用 Fusion 作为推荐效果的衡量指标[7],计算公式如下:

Fusion = \frac{2 \times Precision \times Recall}{Precision + Recall} (12)

3.3 实验结果及分析

3.3.1 选取最优 λ 值

λ 取值分别为 0,0.1,0.2,0.3,……1, 分别计算出相对应的 MAE 值。在 MoviesLens-100k 数据集上在 λ 取值为 0.8 时, MAE 取值最小, 为 0.70。因此 RJpearson 的计算公式为

RJpearson(i, j) = 0.8Rpearson(i, j) + 0.2Jaccard(i, j) (13)

而在数据集 ml-latest small 数据集上最优的 λ 取值为 0.6。

因此在不同数据集进行实验时, 根据计算出的 λ 取值而使用不同的 RJpearson 计算公式。

3.3.2 邻居数量对预测准确性的影响

在进行邻居数量对评分预测准确性影响的实验时, 选取两个不同的评分数据集, 其中一个为数据集 MoviesLens-100k, 另一个数据集是 ml-latest small, 此数据集产生于 2016 年 10 月 17 日, 包含 671 位用户对 9125 部电影的 100004 条评分。

1) 数据集 MoviesLens-100k

邻居数量会对预测的准确性造成影响。选取邻居数量分别为 5,10,20,30,50,100, 在三种算法下分别计算评分预测的 MAE 值, 结果如表 2 所示, 对比图如图 4 所示。

表 2 三种算法对应的 MAE 值

	Pearson	Jaccard	RJpearson
5	0.75798	0.81414	0.68863
10	0.80590	0.81516	0.70004
20	0.83819	0.82209	0.71502
30	0.84986	0.82644	0.72404
50	0.86344	0.83185	0.73476
100	0.86846	0.83961	0.74804

实验结果表明: RJpearson 相似度所对应的 MAE 值明显比 Pearson 相似度和 Jaccard 相似度所对应的 MAE 值小, 改善程度如表 3 所示, 对比图如图 4 所示, 即 RJpearson 相似度明显降低了 User_based CF 的评分预测误差, 提高了预测的准确性。

表 3 RJpearson 相似度对 MAE 值的提升百分比

	Pearson	Jaccard
5	6.94%	12.55%
10	10.59%	11.51%
20	12.32%	10.71%
30	12.58%	10.24%
50	12.87%	9.71%
100	12.04%	9.16%

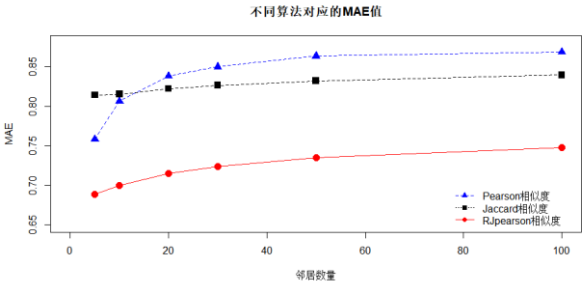


图 4 不同算法对应的 MAE 值

2) 数据集 ml-latest-small

同样选取邻居数量分别为 5,10,20,30,50,100, 在三种算法下分别计算评分预测的 MAE 值, 对比图如图 5 所示。

chinaXiv:201805.00210v1

表 4 三种算法对应的 MAE 值

	Pearson	Jaccard	RJpearson
5	0.43101	0.40762	0.23035
10	0.42537	0.40547	0.22410
20	0.42228	0.40044	0.21780
30	0.42300	0.40150	0.21558
50	0.42081	0.40501	0.21340
100	0.42111	0.40778	0.21116

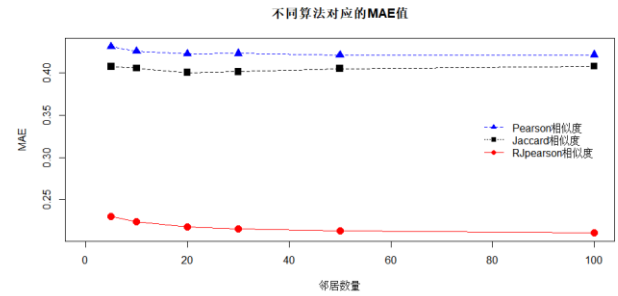


图 5 不同算法对应的 MAE 值

两组实验结果均表明: RJpearson 相似度所对应的 MAE 值明显比 Pearson 相似度和 Jaccard 相似度所对应的 MAE 值小, 而且改善程度均在 10% 以上, 在数据集 ml-latest-small 上, 说明本文算法降低 MAE 值的作用显著, 提高了预测的准确性

3.3.3 推荐数量对推荐结果的影响

根据 3.3.2 节的分析结果可知, Pearson 相似度和 RJpearson 相似度算法均在邻居数为 5 的情况下 MAE 值最低, 表现最优, 因此在邻居数取 5 的基础上, 选择推荐数分别为 5, 10, 15, 20, 25, 分析推荐数对推荐结果的影响, 然后分别产生推荐列表, 计算推荐结果的 Fusion 值, 对比图如图 6 所示。

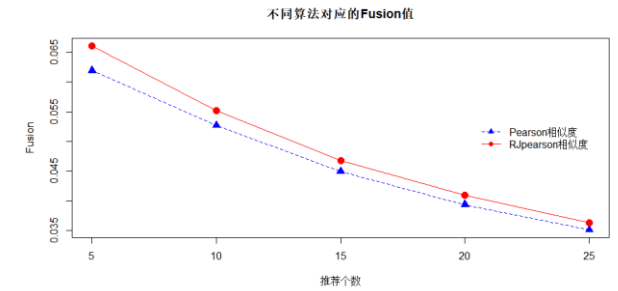


图 6 不同算法对应的 Fusion 值

如图 6 所示, Fusion 值随着推荐数量的增加而降低, 说明推荐结果的准确性随着推荐数量的增加而降低, 因此选择合适的推荐数量至关重要。相对于 Pearson 相似度, RJpearson 相似度相对应的 Fusion 值更高, 说明新的算法提高了推荐的准确性。在推荐数量为 5 时, Fusion 值提升了 0.5%, 虽然 RJpearson 相似度对 Fusion 的提升幅度不大, 但是在一定程度上新算法对 user_based CF 起到了改善作用。

4 结束语

本文主要针对传统的协同过滤算法存在的问题进行改进, 基于 Pearson 相似度引入惩罚因子和 logistic 时间函数对用户评分继续加权处理, 再与 Jaccard 相似度进行结合得到 RJpearson 相似度。在 Movielens 数据集上对改进后的推荐算法进行实验验证, 实验结果表明 RJpearson 相似度准确性更高、降低了评分预测的误差, 从而提高了个性化推荐服务的质量。同时, 本文也存在不足之处, 虽然较为明显的降低了评分预测的误差, 但是推荐结果的精确度和召回率没有较大的改善, 可能是由于数据稀疏造成的, 因此在接下来的工作中将进一步对算法进行优化, 在多个数据集上进行实验验证并会尝试结合文本挖掘或者复杂网络等相关技术对个性化推荐进行研究。

参考文献:

[1] Symeonidis P, Nanopoulos A, Papadopoulos A, et al. Collaborative filtering based on user trends [J]. Advances in Data Analysis, 2006 4425: 375-382.

[2] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions [J]. IEEE Trans on Knowledge and Data Engineering, 2005 17 (6): 734-749.

[3] Zhang Kai, Feng Zhiyong, Chen Shichan, et al. A framework for passengers demand prediction and recommendation [C]// Proc of IEEE International Conference on Services Computing. 2016: 340-347

[4] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of net news [C]// Proc of ACM Conference on Computer Supported Cooperative Work. 1994: 175-186.

[5] Chen Hao, Li Zhongkun, Hu Wei. An improved collaborative recommendation algorithm based on optimized user similarity [J]. Journal of Supercomputing, 2016 72 (7): 2565-2578.

[6] Liu Haifeng, Hu Zheng, Mian A, et al. A new user similarity model to improve the accuracy of collaborative filtering [J]. Knowledge-Based Systems, 2014 (56): 156-166.

[7] Jamali M, Ester M. A random walk model for combining trust based and item-based recommendation [C]// Proc of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2009: 397-406.

[8] Ye HongWu, Dai Yae, Gong SongJie. Combining singular value decomposition and item-based recommender in collaborative filtering [C]// Proc of International Workshop on Knowledge Discovery and Data Mining. 2009: 769-772.

[9] Sarwar M B, Karypis G, A. Konstan J, et al. Application of dimensionality reduction in recommender system-a case study [C]// Proc of ACM Web KDD Workshop, 2000.

[10] Pirasteh P, Jung J J, Hwang D. Item-based collaborative filtering with attribute correlation: a case study on movie recommendation. Intelligent information and database systems [M]. Berlin: Springer International

- Publishing, 2014: 245-252.
- [11] Tsai C, Hung C. Cluster ensembles in collaborative filtering recommendation [J]. Applied Soft Computing, 2012 12 (4): 1417-1425.
- [12] Shaohua W, Zhengde Z, Xin H. The Research on collaborative filtering recommendation algorithm based in improved clustering processing [C]// Proc of IEEE International Conference on Computer and Information Technology. 2015: 1012-1015.
- [13] Lu L, Medo M, Chi H Y, et al. Recommender systems [J]. Physics Reports, 2012 519 (1): 1-49.
- [14] Menaouer B, Atmani B, Matta N. Dynamic knowledge mapping guided by data mining: application on healthcare [J]. Journal of Information Processing Systems, 2013 9 (1): 1-30.
- [15] Motavaselalagh F, Safi Esfahani F, Arabnia R. Knowledge-based adaptable scheduler for SaaS providers in cloud computing [J]. Human-centric Computing and Information Sciences, 2015 5 (1): 1-19.
- [16] Ren Kankan, Qian XueZhong. Research on user similarity measure method in collaborative filtering algorithm [J]. Computer Engineering, 2015. 41 (8): 18-22.
- [17] Su H, Lin X, Yan B, et al. The collaborative filtering algorithm with time weight based on MapReduce [C]// Proc of International Conference on Big Data Computing and Communications. 2015: 386-395.
- [18] Yang L, Hu Y. An improved collaborative filtering algorithm based on the constraint model of confidence [J]. Journal of Computational Information Systems, 2015, 11 (8): 3001-3009.